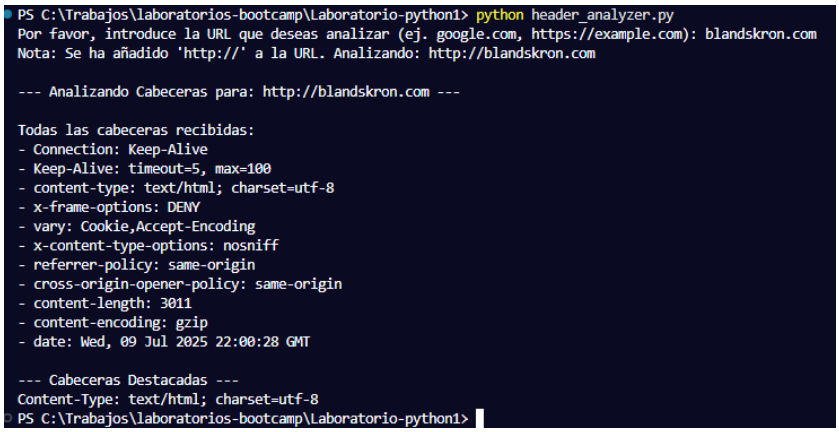
Informe del Ejercicio Práctico: Analizador de Cabeceras HTTP

1. Objetivo del Ejercicio:

Construir y utilizar un script en Python para enviar una solicitud HTTP a un sitio web y mostrar en consola un análisis básico de las cabeceras devueltas por el servidor, obteniendo información útil para el reconocimiento en pruebas de penetración.

Sitio/URL Analizada:

- **URL:** http://blandskron.com



Cabeceras HTTP Detectadas

A continuación, se listan todas las cabeceras HTTP detectadas en la respuesta del servidor:

- **Connection:** Keep-Alive
- **Keep-Alive:** timeout=5, max=100
- **Content-Type:** text/html; charset=utf-8
- **X-Frame-Options:** DENY
- **Vary:** Cookie,Accept-Encoding
- **X-Content-Type-Options:** nosniff
- **Referrer-Policy:** same-origin
- **Cross-Origin-Opener-Policy:** same-origin
- **Content-Length:** 3011
- **Content-Encoding:** gzip
- **Date:** Wed, 09 Jul 2025 22:00:28 GMT

2. Análisis Específico de Seguridad

Se analizaron cabeceras clave para evaluar la postura de seguridad del sitio:

- **Server:**
    - **Estado:** No especificado / Ausente en la respuesta
    - **Observaciones:** Esto es una buena práctica de seguridad, ya que ocultar el tipo y versión del servidor reduce la superficie de ataque y la información disponible para un atacante.
- **X-Powered-By:**
    - **Estado:** No especificado / Ausente en la respuesta
    - **Observaciones:** Similar al header Server, su ausencia es una buena práctica, ya que evita revelar la tecnología backend (PHP, ASP.NET, Node.js, etc.) utilizada en el servidor.
- **Content-Type:**
    - **Estado:** Presente (text/html; charset=utf-8)
    - **Observaciones:** Indica correctamente que el contenido es HTML con codificación UTF-8.
- **Set-Cookie:**
    - **Estado:** No especificado / Ausente en la respuesta
    - **Observaciones:** No se detectaron cookies en la respuesta directa a la petición GET. Si el sitio utiliza autenticación o sesiones, es crucial verificar que las cookies se establezcan con atributos de seguridad como Secure, HttpOnly y SameSite para prevenir ataques.
- **Strict-Transport-Security (HSTS):**
    - **Estado:** Ausente en la respuesta
    - **Observaciones:** La ausencia de esta cabecera es una vulnerabilidad, ya que permite ataques tipo downgrade y ataques Man-in-the-Middle (MITM). Los navegadores no recordarán que el sitio debe usarse siempre por HTTPS, incluso si se accede a él por esta vía.
- **X-Frame-Options:**
    - **Estado:** Presente (DENY)
    - **Observaciones:** Esta es una buena práctica de seguridad. La directiva DENY previene que el sitio sea incrustado en <iframe>, <object>, <embed>, lo que protege contra ataques de clickjacking.

- **X-Content-Type-Options:**
    - **Estado:** Presente (nosniff)
    - **Observaciones:** Otra buena práctica de seguridad. La directiva nosniff evita que los navegadores "adivinen" o interpreten incorrectamente el tipo MIME de los archivos, lo que puede prevenir ataques basados en inyección de contenido.
- **Referrer-Policy:**
    - **Estado:** Presente (same-origin)
    - **Observaciones:** Buena práctica. Limita la información que se envía en la cabecera Referer solo a solicitudes que se originan en el mismo sitio, reduciendo la fuga de información sensible.
- **Cross-Origin-Opener-Policy:**
    - **Estado:** Presente (same-origin)
    - **Observaciones:** Buena práctica que mejora el aislamiento entre pestañas/ventanas del navegador. Ayuda a mitigar ataques tipo Spectre y protege contra la fuga de información a ventanas pop-up maliciosas.

3. Vulnerabilidades / Riesgos Detectados

Basado en el análisis de cabeceras, se identificaron los siguientes riesgos de seguridad:

1. **No fuerza HTTPS:**
    - El acceso se realizó por HTTP (http://blandskron.com) y, por la ausencia de un redireccionamiento automático y la falta de HSTS, se expone al sitio a ataques de interceptación (Man-in-the-Middle) si un usuario intenta acceder por HTTP.
2. **Falta de Strict-Transport-Security (HSTS):**
    - Esta cabecera es crucial para garantizar que los navegadores utilicen siempre HTTPS una vez que han accedido al sitio de forma segura. Su ausencia permite ataques de degradación de protocolo (downgrade attacks).

4. Buenas Prácticas Detectadas

El sitio blandskron.com muestra las siguientes buenas prácticas de seguridad en sus cabeceras:

- **Ocultamiento de información del servidor y tecnología (Server, X-Powered-By):** Su ausencia es una buena práctica que reduce la información disponible para un atacante sobre la infraestructura del servidor.
- **Implementación de X-Frame-Options: DENY:** Protege eficazmente contra ataques de clickjacking, impidiendo que el sitio sea incrustado en frames de otros dominios.
- **Implementación de X-Content-Type-Options: nosniff:** Previene que el navegador "adivine" el tipo de contenido, mitigando ataques de inyección de contenido.
- **Uso de Referrer-Policy: same-origin:** Limita la cantidad de información del Referer que se envía a otros dominios, mejorando la privacidad y seguridad.
- **Uso de Cross-Origin-Opener-Policy: same-origin:** Mejora el aislamiento de seguridad para la página, protegiendo contra posibles ataques de filtración de información y mitigando riesgos como Spectre.

5. Recomendaciones Finales

Basado en el análisis, se recomiendan las siguientes acciones para mejorar la seguridad del sitio:

1. **Forzar HTTPS automáticamente:** Configurar el servidor web (Apache, NGINX, Cloudflare, etc.) para redirigir automáticamente todo el tráfico HTTP a HTTPS.
2. **Agregar la cabecera Strict-Transport-Security (HSTS):** Implementar esta cabecera en las respuestas HTTPS con una duración adecuada (ej., Strict-Transport-Security: max-age=31536000; includeSubDomains; preload) para asegurar que los navegadores siempre usen HTTPS.
3. **Evaluar la seguridad de las cookies (si aplica):** Si el sitio utiliza cookies para sesiones o cualquier otra función que maneje datos sensibles, asegurar que tengan los atributos Secure, HttpOnly, y SameSite=Strict o Lax para protegerlas de ataques como XSS y CSRF.
4. **Auditar cabeceras dinámicas:** Revisar las cabeceras que puedan aparecer solo en rutas específicas (ej., login, formularios, paneles de administración) para asegurar que también cumplan con las mejores prácticas de seguridad.
5. **Realizar escaneos de seguridad automatizados:** Utilizar herramientas como Security Headers.com o OWASP ZAP para realizar revisiones automáticas y continuas de las cabeceras y otras vulnerabilidades web.